

## PATIENTS:

In the relation we have ID as the primary key that can uniquely define all the attributes. Additionally [ first name, last name, dob, email] is a candidate key that also defines the tuples uniquely. However, this relation is not BCNF because there exists this FD {address\_country,address\_zip→ address\_state } which violates the BCNF rule since the LHS (address\_country,address\_zip) is not a super key. To normalize this relation we can generate two relations as follows:

**Patients**={id, password, first\_name, last\_name, dob, phone, email, address\_street, address\_country, address\_zip, insurance, pharmacy\_address }

**Address\_info**={ address\_country, address\_zip, address\_state}

In the second relation above, (**address\_zip, address\_state**) pair is the primary key and there is no other violating FD in this one. therefore this is also in BCNF form.

## PRESCRIPTION:

The relation contains only one key, id, which uniquely determines a specific prescription that a doctor has given to a patient. It violates BCNF since appointment\_id functionally determines patient\_id and doctor\_id but appointment\_id is not a key. That means you have to decompose it into (id, name, dosage, expiration, appointment\_id) and (appointment\_id, patient\_id, doctor\_id). Since the second relation is already part of the appointment relation, we can get rid of it entirely and are left with (id, name, dosage, expiration, appointment\_id). The only functional dependency is id -> {name, dosage, expiration, appointment\_id} and id is key so it is in BCNF form.

## APPOINTMENT:

This relation contains only one key: id, the primary key used to identify appointments within the system. The other attributes might not be unique: it is entirely possible that multiple appointments will start or end at the same time, and that a doctor might leave the same appointment notes for multiple appointments (example: "Everything looks good!"), and patients and doctors are likely to be involved in multiple appointments each, meaning patient\_id and doctor\_id can't be unique.

id functionally determines all other attributes by definition. There are no other functional dependencies in this relation. Since appointments can be different durations, there is no guarantee that two appointments that start at the same time will end at the same time. Doctors are likely to leave different notes on different appointments, and patients will probably receive different notes at every appointment. Since in every nontrivial functional dependency, the left-hand side is a superkey, **APPOINTMENT** is already in BCNF.

## DOCTOR:

This relation contains three keys: id, the primary key used to identify doctors within the system, phone, a unique key that allows people to contact a specific doctor, and email, another piece of contact information that doubles as a way to log in to the online portal. Since people will use

phone numbers and emails to get in touch with specific doctors, it is essential that no two doctors share a phone number or email address.

The other attributes of **DOCTOR** might not be unique. For instance — although unlikely, it is possible that two doctors with the exact same names work together at the hospital! A much more likely example: most departments will have several doctors working in them.

BCNF requires that in every functional dependency, the left-hand side is a superkey. In **DOCTOR**, all three keys determine all other attributes by definition. No other functional dependencies exist, meaning that **DOCTOR** is already in BCNF.

#### **DEPARTMENT:**

There are two keys in this relation: **id**, the primary key used to identify departments within the system, and **name**, a unique key used by doctors and patients to identify and categorize hospital operations. There is only one other attribute, **building**, which is not a key, because multiple departments may operate in the same physical building. It is obvious, then, that **building** does not functionally determine any other attributes. Both keys functionally determine everything else, but this is consistent with the requirements of BCNF: in every nontrivial functional dependency, the left-hand side is a superkey. In this case, the LHS is **id** or **name** or {**id**, **name**}, all of which are superkeys. This means that **DEPARTMENT** is already in BCNF.